

# 多重評価方式を備えた並列Lispシステムに関する研究

著者	川本 真一
号	95
発行年	1998
URL	<a href="http://hdl.handle.net/10097/12809">http://hdl.handle.net/10097/12809</a>

氏 名 (本 籍)	川 本 眞 一 (東京都)
学 位 の 種 類	博 士 (情報科学)
学 位 記 番 号	情 博 第 95 号
学 位 授 与 年 月 日	平 成 10 年 7 月 9 日
学 位 授 与 の 要 件	学位規則第 4 条第 1 項該当
研 究 科, 専 攻	東北大学大学院情報科学研究科 (博士課程) 情報基礎科学専攻
学 位 論 文 題 目	多重評価方式を備えた並列 Lisp システムに関する研究
論 文 審 査 委 員	(主査) 東北大学教授 伊 藤 貴 康    東北大学教授 丸 岡 章 東北大学教授 白 鳥 則 郎    東北大学教授 川 又 政 征 (工学研究科)

## 論 文 内 容 要 旨

### 第 1 章 序 論

1 章は序論である。プログラムの高速化の手法の 1 つに並列処理がある。数値計算の分野で扱うデータは規則正しい配列であるため、数値計算プログラムを自動的にかつ効率的に並列化することは容易であり、数値計算向けの Fortran や C 言語における並列処理は、自動並列化コンパイラのアプローチが取られる。一方、人工知能や記号処理の分野で用いられる Lisp 言語では、Lisp プログラムの関数型言語の部分についての並列化は容易であるが、プログラムの再帰性や扱うデータが不規則であることから、効率的な自動並列化は困難であり、Lisp 言語における並列処理は、並列 Lisp 言語を用いて並列プログラミングを行うアプローチが取られる。このアプローチによって効率的な並列処理を行うには、性能の良い並列 Lisp 言語の処理系の実現が必要となる。

並列 Lisp 言語における並列構文の評価法として、従来は ETC (Eager Task Creation) と呼ばれる方法が用いられてきた。この方法によって並列プログラムを評価すると、プロセスが過剰に生成され実行効率が悪いというプロセス過剰生成問題が発生する。これに対して、並列 Lisp の 1 つである Multilisp[1] の処理系 Mul-T において、LTC (Lazy Task Creation)[5] と呼ばれるプロセス生成を抑制できる方法が提案されている。しかし、この方法は、並列構文として future 構文しか考えられていない。文献[3]において、スティール評価法という、あらゆる並列構文に適用できプロセス生成を抑制できる効率の良い評価法が提案されている。

本研究は、並列 Lisp 言語 PaiLisp[2] の効率の良い処理系の実現を目的とし、まず、共有メモリ型並列処理システムの下での効率の良い処理系の構成法を示した。そしてこの構成法に基づき、ETC、スティール評価法、LTC の 3 つの並列 Lisp の評価法を実装し、問題の性質に応じて評価法が選択可能な多重評価法を備えた並列 Lisp システムを実現した。そして、このシステムを用いて評価実験を行い、スティール評価法が最も効率の良い評価法であることを実験的に示した。

### 第 2 章 並列 Lisp 言語 PaiLisp

2 章では、並列 Lisp 言語 PaiLisp[2] と、PaiLisp プログラムの実行上の問題について述べている。

PaiLisp は、共有メモリ型並列計算機のための Scheme をベース言語とした並列 Lisp 言語であり、pcall, pbegin,

pif, future などの豊富な並列構文を備えている。本章の前半では、PaiLisp の構文と主要な並列構文の意味について述べている。

PaiLisp には逐次構文に対応した並列構文があり、逐次プログラムの逐次構文に対応する並列構文に置き換えることによって、容易に並列化できる。しかし、並列化したプログラムの実行という観点からは次のような問題がある。従来から用いられてきた並列構文の評価法である ETC は、並列構文に遭遇すると必ずプロセスを強制的に生成するので、再帰的に並列構文を呼び出すプログラムを ETC によって評価すると、プロセスが過剰に生成され実行効率が悪くなるプロセス過剰生成問題が発生する。プロセスの過剰な生成を抑制できる評価法として、スティール評価法が提案されている[3]。本章の後半では、ETC のプロセス過剰生成問題について説明し、スティール評価法を用いれば、プロセス生成を抑制して効率的な並列実行ができることを例を用いて説明している。

### 第3章 マルチスレッド方式による PaiLisp インタプリタの実現

3 章では、共有メモリ型並列処理システムの下でのマルチスレッド方式を用いた効率の良い処理系の構成法を与え、この構成法と並列構文の評価法である ETC に基づいた PaiLisp のインタプリタ PaiLisp/MT (ETC) の実現について述べている。

共有メモリ型並列処理システムの下で効率の良い処理系を実現するには、並列に実行される複数の計算が、共有メモリへ効率的にアクセスできる必要がある。共有メモリへのアクセスを効率良くかつ汎用的に実現するため、マルチスレッド方式を用い、1つのタスク内に共有されるデータを配置する。そして、各計算をスレッドに割り当てて、それらを実行するスレッドをタスク内のデータに直接アクセスさせる。この構成法に基づき、並列構文の評価法として ETC を用いた PaiLisp のインタプリタ PaiLisp/MT (ETC) を実現した。各スレッドには PaiLisp インタプリタを実現する仮想計算機を割り当て、それらの各仮想計算機を直接タスク内の共有データにアクセスさせているため、共有メモリへのアクセスが効率的に行われる。

従来の UNIX OS による実装では、OS の提供する共有メモリ機構においてメモリへのアクセスコストが大きく効率が悪くない。一方、アセンブラ言語による実装の場合は、共有メモリに対するアクセスは高速であるが、実装はマシンに依存し、汎用性が無くなる。これらの方法に比べて、本章で与えた方法は、共有メモリ型並列処理システムの下での処理系を効率良くかつ汎用的に実現できる方法になっている。

本章の後半では、上記の構成法に基づいて実現された PaiLisp インタプリタ PaiLisp/MT (ETC) における ETC に基づく並列構文の実装の詳細について述べている。

### 第4章 スティール評価法に基づく PaiLisp インタプリタの実現

4 章では、3 章で実現した ETC に基づく PaiLisp のインタプリタ PaiLisp/MT (ETC) のプロセス過剰生成問題を解決することができる、スティール評価法に基づく効率の良い PaiLisp インタプリタの実現法について述べている。

文献[3]で提案されたスティール評価法は次のような評価法である。 $(\text{pcall } f e_1 \cdots e_n)$  のような並列構文を含む式を評価したプロセッサは、その時点では式  $e_1, \dots, e_n$  を評価するプロセスを生成せず、式  $e_1, e_2, \dots$  を逐次的に評価していく。他のプロセッサが空き状態になると、そのプロセッサは式  $e_1, e_2, \dots$  を評価しているプロセッサから式を  $e_n, e_{n-1}, \dots$  の順に1つ取って、それを評価するプロセスを生成し実行する。このように、スティール評価法では、プロセッサが空き状態になったときにのみプロセス生成が行われるため、プロセス生成を抑制でき、効率的な並列実行を行うことができる。

本章では、PaiLisp の構造化並列構文のスティール評価法による実現法を与え、それをリンク構造とポインタを用いて効率良く実装する方法を与えた。また、**future** 構文のスティール評価法と LTC[5] による実現法を与えた。これらの方法に基づき、スティール評価法と LTC を実装し、ETC、スティール評価法、LTC(**future** の構文のみ) の3つの評価法からなる多重評価法を備えた PaiLisp インタプリタ PaiLisp/MT を実現した。そして、PaiLisp/MT における同一の環境下で実現された ETC、スティール評価法、LTC の3つの評価法の比較を行い、スティール評価法が最も優れた評価法であることを示した。

## 第5章 PaiLisp のプログラミング環境

5章では、PaiLispのプログラミング環境に関して筆者が行った研究について述べている。

PaiLisp インタプリタよりもさらに効率的に PaiLisp プログラムを実行するという観点から、PaiLisp コンパイラが文献[4]において試作されている。このコンパイラについて、並列構文に対するコード生成法を改良してより効率の良いコード生成を行えるようにし、また **future** 構文に対するコード生成が行える拡張を行った。

効率的な並列プログラムの作成を支援するという観点から、PaiLisp のためのパフォーマンスデバッガとして使用できるプロセス視覚化ツール Process Watcher を試作し、また、逐次 Scheme プログラムを自動並列化する簡便な自動並列化システムである CAP システムの試作を行った。

## 第6章 PaiLisp システムの実験評価と応用

6章では、多重評価法を備えた PaiLisp/MT システムの実験評価について述べている。

共有メモリ型並列計算機 DEC7000 上に実現された PaiLisp/MT システムを用いて評価実験を行った。まず、インタプリタの場合について、各種のコストの計測を行い、ベンチマークプログラムを用いて ETC, スティール評価法, LTC の比較実験を行った。ついで、コンパイラの場合について、各種のコストの計測を行い、ベンチマークプログラムを用いて ETC とスティール評価法の比較実験を行い、また、コード生成法を改良したスティール評価法についての実験を行った。さらに、実用的な規模と性質を持った応用プログラムの例として逐次 Scheme インタプリタとペトリネットの解析ツールを取り、これらのプログラムを並列化し評価実験を行った。これらの評価実験の結果によって、インタプリタにおいてもコンパイラにおいてもスティール評価法が最も効率が良いことを実験的に示した。

## 第7章 結 言

7章は結言であり、本研究の主要な結果と、今後の課題について述べている。

まず、マルチスレッド方式に基づき、並列計算機の共有メモリへのアクセスを効率的に実現できる処理系の実装法を与え、この実装法によって PaiLisp の効率の良いインタプリタを実現した。また、このインタプリタの下で、スティール評価法と LTC に基づく並列構文の実装法を与え、多重評価法を備えた PaiLisp システムを実現した。ベンチマークプログラムおよび応用プログラムを用いた評価実験によって、ETC および LTC に比べてスティール評価法が最も優れた評価法であることを実験的に示した。

## 参 考 文 献

- [1] R.Halstead,Jr.: Multilisp: A language for concurrent symbolic computation, ACM Trans. On Programming Languages and Systems, Vol.4, No.7, pp.501-538 (1985).
- [2] T.Ito, M.Matsui: A parallel Lisp language PaiLisp and its kernel specification, LNCS, Vol.441, pp.58-100, Springer (1990).
- [3] T.Ito: Efficient evaluation strategies for structured concurrency constructs in parallel Scheme systems, LNCS, Vol.1068, pp.22-52, Springer (1996).
- [4] T.Ito, S.Kawamoto, and M.Umehara: A Multi-threaded implementation of PaiLisp interpreter and compiler using the steal-help evaluation strategy, Advanced Lisp Technology, Advanced Information Processing Technology, Vol.6, Gordon and Breach Publishing Company (1999年出版予定).
- [5] E.Mohr, D.A.Kranz, R.Halstead,Jr.: Lazy task creation: A technique for increasing the granularity of parallel programs, IEEE Trans. Parallel and Distributed systems, Vol.2, No.3, pp.264-280 (1991).

## 論文審査の結果の要旨

人工知能や記号処理の分野における複雑で不規則な構造をしたデータの対話型処理や実時間処理の要求に応えるため、並列 Lisp 言語の高性能並列処理系が必要とされている。

著者は、並列 Lisp 言語に対する代表的な並列評価法を効率のよい単一の並列計算機構の上に実現し、問題の性質に応じて評価方式が選択可能な多重評価方式を備えた並列 Lisp システムに関する研究を行った。本論文はその成果をとりまとめたもので、全文 7 章よりなる。

第 1 章は序論である。

第 2 章では、並列 Lisp 言語 PaiLisp と高性能並列言語処理系実現上の課題について論じている。

第 3 章では、マルチスレッド方式による並列計算機構の構成法を与え、それを基に、並列構文に遭遇すると必ずタスク生成を行う評価法である ETC (Eager Task Creation) によって実現した PaiLisp インタプリタの詳細を与えている。ここで与えた並列計算機構の構成法は、効率のよい並列計算機構を作成する上で有用な知見を与えるものである。

第 4 章では、プロセス過剰生成を抑制できるスティール評価法と遅延タスク生成法に基づく PaiLisp インタプリタの効率のよい実現法を与えている。この方法を用いて実現したスティール評価法と遅延タスク生成法を ETC と比較し、スティール評価法が最も優れた並列評価法であることを明らかにしている。これは効率のよい単一の並列計算機構の上で多重評価方式を備えた並列処理系を実現し、同じ計算環境のもとで評価方式を比較することによって得られた重要な成果である。

第 5 章では、PaiLisp インタプリタをより高性能なシステムとするためのコンパイラ機能、プロセス視覚化機能を用いたパフォーマンスデバッガ、および、逐次 Lisp プログラムの簡便な自動並列化機能を実現している。

第 6 章では、PaiLisp システムの基本的な性能指標を計測し、ベンチマークプログラムおよび応用例についての実験を行っている。これらの実験によって、スティール評価法がインタプリタ実行のみならずコンパイラ実行においても最も優れた評価法であることを示している。これは実用上有用な成果である。

第 7 章は結言である。

以上要するに本論文は、高性能並列言語処理系の構成法を与え、マルチスレッド方式による並列計算機構を用いて多重評価方式を備えた並列 Lisp システムを実現したものであり、情報基礎科学の発展に寄与するところが少なくない。

よって、本論文は博士（情報科学）の学位論文として合格と認める。